UNITED STATES PATENT APPLICATION

OF

THOMAS DRIEMEYER AND ROLF HERKEN

FOR

SCALABLE, MULTI-USER SERVER AND METHOD FOR RENDERING IMAGES FROM
INTERACTIVELY CUSTOMIZABLE SCENE INFORMATION

Certificate of Express Mailing

Express Mail Mailing Label No. EK 904 503 626 US

Date of Deposit April 28, 2001

I hereby certify that this paper or fee is being deposited
with the United States Postal Service "Express Mail Post Office
To Addressee" Service under 37 CFR 1.10 on the date indicated
above and is addressed to the Commissioner of Patents and
Trademarks, Washington, D. C. 20231.

By _____

Richard A. Jordan

1   **FIELD OF THE INVENTION**

2   The invention relates generally to the fields of computer graphics and distribution of

3   information in graphical form, generally in the form of rendered images, over networks such as the

4   Internet. The invention provides a new and improved scalable, multi-user server and method for

5   rendering images from interactively customizable three-dimensional scene information.

6   **BACKGROUND OF THE INVENTION**

7   Devices, such as computers, personal digital assistants (PDA's), cellular telephones, and the

8   like, that can generate, process, display and otherwise make use of information in digital form, are

9   often connected into networks to facilitate sharing of information thereamong. In some networks,

10 so-called local area networks (LAN's), the networks extend over a relatively small geographic

11 region, such as a building or group of buildings. Other networks, so-called wide area networks

12 (WAN's), the networks extend over larger geographical regions, and may include LAN's as parts

13 thereof. Some networks are private, maintained by an organization such as a corporation,

14 government agency and the like, and may be accessed only by, for example, employees and other

15 authorized people. On the other hand, some networks, such as the Internet or World Wide Web, are

16 public and typically may be accessed by anyone who has access to a suitable digital device and

17 network connection.

18   A number of types of paradigms and protocols exist for transferring information over a

19 network, such as a WAN such as the Internet and World Wide Web (generally, "Internet"), or a LAN

20 ("Intranet").. One paradigm is the so-called client/server paradigm, in which some devices, which

21 are referred to as servers, store digital information that may be retrieved by other devices, which are

22 referred to as clients. Several protocols exist for retrieving information, including the so-called file-

23 transfer protocol ("FTP") for facilitating the retrieval of individual information files for, for example,

24 later processing,, and the HyperText transfer protocol ("HTTP") for facilitating the retrieval of one

1   or more information files, at least one of which will be in the so-called HyperText Markup Language

2   ("HTML"), all of which constitute a Web page. Typically, Web pages include textual and graphical

3   information that is to be displayed on a display provided by the client device.

4   One popular type of program that is often used for retrieving and using information files

5   comprising a Web page is referred to as a browser. A browser provides a convenient mechanism

6   by which a user can identify the particular item of information that is to be downloaded, by

7   providing a "URL," or "universal resource locator." A URL is identifies a computer, network

8   domain or Web site (generally, "web site") from which the item of information is to be retrieved, and

9   may also specify a particular item of information that is to be retrieved. Typically, URL's are in

10   relatively user-friendly form, typically identifying at least the Web site by name or a mnemonic of

11   the name of the person or organization that maintains the Web site. The browser will convert at least

12   the portion of the URL that identifies the web site to a network address, which is typically in

13   numerical form, which it uses to contact the Web site and establish a "connection" therewith. A

14   browser may need to contact another device, referred to as a name server, that maintains a

15   concordance between URL's and network addresses, to obtain the network address. After the

16   browser has the web site's network address, it can use the network address, the identification of the

17   particular item of information that is to be retrieved, and possibly other parameters to establish a

18   connection with the Web site and initiate retrieval of the information item.

19   A browser typically retrieves information in the form of documents or "Web pages," which

20   may include text and graphical images, and may also include streaming video and audio information.

21   The textual information is specified in one of a number of document description languages, typically

22   in the well-known HyperText Markup Language (HTML). If a Web page is to have one or more

23   graphical images and/or video information displayed therewith, the HTML description identifies the

24   locations on the Web page at which the images or streaming video information are to be displayed

25   and the sizes of regions of the Web page on which the respective images or video information are

26   to be displayed. In addition, the HTML description will provide URL's for the respective images

27   and streaming video information. Similarly, if the Web page is to be displayed along with audio

28   information, the HTML description will specify the audio information that is to be played.

-2-

1   As the browser displays the Web page on the computer's video display screen, it will display

2   the text as specified in the HTML description, in the process reserving regions of the displayed Web

3   page on which the respective images are to be displayed. In addition, the browser will retrieve the

4   graphical images, using the provided URL's provided in the HTML description in a manner similar

5   to that described above, and display them in the regions on the video display screen that were

6   reserved therefor. Furthermore, if streaming video information is to be displayed, the browser can

7   initiate retrieval of the streaming video information either while displaying the other elements of the

8   Web page or at some point after the Web page has been displayed. The user may need to perform

9   some action, such as actuating a pushbutton displayed on the Web page. A pushbutton can be

10   actuated in any of a number of ways, including clicking on it using a pointing device such as, for

11   example, a mouse, pressing on the region of a touch screen on which the pushbutton is displayed by,

12   for example, a stylus, or any other mechanism for actuating a pushbutton displayed on a video

13   display screen as will be appreciated by those skilled in the art. Audio information may be retrieved

14   in a manner similar to the streaming video information and played through an audio reproduction

15   device, such as a speaker, provided with the computer.

16   In addition to text, image, streaming video and audio information, a Web page may also be

17   associated with programs, termed "applets," that may be retrieved with the other types of information

18   and executed under control of the browser.

19   Generally, the Web pages that are currently displayed by browsers are static documents. That

20   is, a user, through the browser, requests a Web page, and the browser retrieves the information

21   associated with the Web page and displays it. Essentially, when the Web site has provided the

22   information associated with the Web page, that essentially ends the transaction between the browser

23   and the Web site in relation to that Web page. If the user wishes to retrieve another Web page from

24   the same Web site, he or she may do so by, for example, entering another URL or actuating a link

25   on the Web page that is currently being displayed, which will initiate another transaction.

26   Typically, a user cannot modify or customize the way a Web page is displayed, unless an

27   image depicts a scene that is to be displayed in three-dimensional form, in, for example VRML or

-3-

1   X3D format. For such images, by actuating controls that may be provided on the Web page, a user

2   can enable the three-dimensional scene to be displayed from a number of orientations. While this

3   can be useful in some situations, there are a number of limitations that make it less than optimum.

4   For example, the amount of information required to define objects in a three-dimensional scene in

5   any significant degree of detail can be quite large, and, given bandwidth limitations that are typical

6   in many connections to the World Wide Web, it would require an inordinate amount of time to

7   retrieve the information required to display the three-dimensional scene if the scene has any

8   significant degree of detail. Accordingly, typically for three dimensional scenes, the amount of

9   image information will be limited sufficiently so that the three-dimensional scenes have only a few

10 relatively small objects and textures, with an extremely limited range of illumination and surface

11 property effects. In addition, although a user can change the viewpoint from which the scene is

12 displayed, he or she cannot change the orientation or a number of other characteristics of the objects

13 in the scene.

14   Even if the bandwidth were sufficient to enable sufficient three-dimensional scene

15 information to be retrieved within a reasonable amount of time to facilitate display of the scene with

16 a more photo-realistic quality, in a number of situations it may be undesirable to transfer the

17 information to the user. For example, if a manufacturer uses the Web site to provide information

18 about its products for, for example, potential customers, it may not wish to make information

19 sufficient to provide photo-realistic three-dimensional images available for retrieval, since

20 information that is sufficiently detailed to generate such images may also be sufficiently detailed to

21 provide a significant amount of design information that may be of interest to competitors. This is

22 particularly the case if the information is sufficiently detailed to allow a user to modify or customize

23 the scene. For example, if the manufacturer is an automobile manufacturer, it may be desirable to

24 allow a user to not only view the automobile from user-selectable orientation, but also to modify or

25 customize the scene, by, for example, changing the color and texture of various surfaces, changing

26 the positions of light sources, enable the automobile to be displayed with doors, hood and/or trunk

27 in an open position, and the like. The amount of information that would be necessary to allow a user

28 to perform such operations may require a significant amount of time to transfer. In addition, the

1 amount of information that may be required may constitute a significant amount of the design

2 information for the object(s) in the scene, which may be confidential.

3   Accordingly, it will be desirable to maintain the three-dimensional scene information on the

4 Web site and have the Web site render two-dimensional images in orientations and with

5 modifications and customizations of the scene as specified by the user, and transmit the two-

6 dimensional image information to the user's browser for display. However, problems arise since not

7 only will the Web site need to retrieve the information from databases on which the information is

8 stored for transmission to the user's browser, but need also render the two-dimensional images from

9 orientations and with modifications and customizations specified by the user. For example, if a

10 number of users are accessing the Web site concurrently, the amount of processing power required

11 to render the images in a reasonable amount of time can become quite large. In addition, problems

12 can arise if a group of users are making use of the same scene, for whom customizations made by

13 any of the users in the group are to be incorporated into the scene as used by all of the members of

14 the group, since all of the customizations would need to be transmitted to all of the users and

15 incorporated into their respective three-dimensional scenes.

SUMMARY OF THE INVENTION

17   The invention provides a new and improved scalable, multi-user server and method for

18 rendering images from interactively customizable three-dimensional scene information.

19   In brief summary, the invention provides a server for use in connection with a network

20 including at least one client and a communication link interconnecting the client and server. The

21 server comprises an image rendering module and an interface. The image rendering module is

22 configured to render, from three-dimensional scene data representing a scene, a two-dimensional

23 image. The interface configured to transmit the two-dimensional image over the communication link

24 to the client.

1   The server is also provided with a user interaction control module that regulates interactions

2   between the server, in particular the image rendering module, and respective clients who may be

3   using the server concurrently to control images in which customizations requested by, for example,

4   respective clients are rendered.

5   **BRIEF DESCRIPTION OF THE DRAWINGS**

6   This invention is pointed out with particularity in the appended claims. The above and

7   further advantages of this invention may be better understood by referring to the following

8   description taken in conjunction with the accompanying drawings, in which:

9   FIG. 1 is a functional block diagram of an arrangement including a scalable multi-user server

10   that provides for rendering of images based on scenes that can be interactively customized by clients,

11   constructed in accordance with the invention;

12   FIG. 2 is a functional block diagram of the server depicted in FIG. 1; and

13   FIG. 3 is a flow diagram useful in understanding operations performed by a user manager

14   in the server depicted in FIG. 2.

15   **DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT**

16   FIG. 1 is a functional block diagram of an arrangement 10 including a scalable, multi-user

17   server 11 that provides for rendering of images based on scenes that can be interactively customized

18   by clients, constructed in accordance with the invention. The server 11 provides web pages,

19   including text and images, individual images, sequences of images, streams of images that provide

20   for the perception of continuous motion of rendered scene elements (generally, "streaming video"),

21   and the like to the respective clients. With reference to FIG. 1, in addition to the server 11,

22   arrangement 10 includes a plurality of client devices 12A,...,12N (generally identified by reference

23   numeral 12n) that can access the server 11 over a network 13. In one embodiment, the network 13

1   is a wide area network (WAN) such as the Internet and World Wide Web, but it will be appreciated

2   that an arrangement in accordance with the invention can include any form of network, including

3   local area networks (LAN's). The client devices 11n may be any kind of information utilization

4   devices that may receive, utilize and display information in digital form, including computers such

5   as personal computers, workstations, personal digital assistants (PDA's), cellular telephones and the

6   like. The server 11 can be implemented using, for example, a suitably-programmed computer.

7   Generally, the client devices 12n and server 11 communicate over the network 13 according

8   to a client/server communication model. According to that model, a client, such as a client device

9   12n, generates an information retrieval request that requests retrieval of a particular item or items

10   of information from the server 11, and transmits it to a server, such as the server 11, over the

11   network 13. The information retrieval request may be generated in response to input provided by

12   an operator, in response to a request generated by a program, or in response to other occurrences as

13   will be apparent to those skilled in the art. When the server receives an information retrieval request

14   from a client, depending on the information whose retrieval is being requested, it may obtain the

15   information item(s) from, for example, a database that it maintains, and transmit them to the client

16   over the network 13. When the client receives the information item(s), it can make use of the item(s)

17   in any of a number of ways. If, for example, the information item(s) comprise a Web page, the client

18   device can display the Web page on a display device, store the Web page in a storage device, provide

19   the web page through a suitable editor to a user to allow him or her to edit the Web page and so

20   forth. The uses to which a client device 12n may put other types of information item(s) will be

21   apparent to those skilled in the art. If a Web page comprises several components, including one or

22   more textual components, images and the like, a client device 12n may need to generate multiple

23   information retrieval requests each requesting retrieval of one or more of the Web page's

24   components. All of the information retrieval requests for the various components of the Web page

25   may be transmitted to the same server, such as server 11, for response. On the other hand, one or

26   more of the information retrieval requests for various ones of the components of the Web page may

27   be transmitted to other servers (not shown), which can provide the respective components. As the

28   respective client 12n that issues the information retrieval request(s) receives the requested

1   information, or at some point thereafter, it can make use of the information, and, if the requested

2   information is a Web page, display the Web page.

3   The invention provides an arrangement whereby a server, such as the server 11, can provide

4   Web pages to client devices 12n, which Web pages contain two-dimensional images that are

5   rendered from three-dimensional scenes. The invention further provides an arrangement whereby

6   a server, such as the server 11, can provide images rendered from three-dimensional scenes, which

7   scenes can, in turn, be interactively modified or customized (generally "customized") during a

8   session in response to customization input from a user who is using a respective client device 12n',

9   or a user who is another client device 12n". A number of clients 12n', 12n",... may request images

10   of, for example, the same scene contemporaneously, from the same or different viewing directions,

11   and the server 11 can efficiently render the images and transfer them to the respective clients 12n',

12   12n" for use thereby. A client device 12n' can request customizations to the scene, and the server

13   11 can selectively enable the requested customizations to be depicted only in images, sequences of

14   images, or images comprising streaming video, that are rendered for that client device 12n'.

15   Alternatively, the server 11 can selectively enable the customizations requested by one client device

16   12n' to be depicted in images, sequences of images, or images comprising streaming video, that are

17   rendered for selected ones of the client devices 12n', 12n",... that are contemporaneously engaged

18   in sessions involving the same scene. As a further alternative, the server 11 can enable the

19   customizations requested by one client device 12n' to be depicted in images, sequences of images,

20   or images comprising streaming video, that are rendered for all of the client devices 12n', 12n",... that

21   are contemporaneously engaged in sessions involving the same scene. In all cases, the images

22   provided by the server 11 may be still images, sequences of images, streaming video, or any other

23   form or arrangement by which images can be provided to a client device.

24   Generally, during a session, the user will initially initiate retrieval of a Web page from the

25   server 11, which Web page will include an image. The Web page also provides a set of tools, which

26   can be displayed as, for example, push buttons, dial objects, radio buttons, dialog boxes and the like

27   as part of the Web page. During the session, the user, using user input devices provided by his or

28   her client device 12n', can manipulate the tools to enable customizations to be made to the scene,

-8-

1   customizations to be made the viewing direction, and/or types of customizations as will be apparent

2   to those skilled in the art, which can be made to a scene and how an image of the scene is rendered.

3   The client device 12n' can transmit indicia indicating the customizations that were requested by the

4   user to the server 11, which, in turn, can generate a new image reflecting the customizations, and

5   transmit the new image to the user's client device. As the client device 12n' receives the new image,

6   or at some point thereafter, it can substitute the new image for the previous image in the Web page.

7   These operations can be repeated during a session in response to user customization requests.

8   Similarly, if the customizations requested by a user using one client device 12n' are to be depicted

9   in images rendered for other client devices 12n",... contemporaneously engaged in a session

10   involving the same scene, the server 11 can render new images depicting the customizations and

11   transmit them to the other client devices 12n",... for display. In addition, particularly in the case of

12   image sequences, streaming video, or the like, the server 11 can provide sequences of images or

13   streaming video to a respective clients without requiring user customization requests or other

14   requests therefrom using a so-called "push" methodology.

15   A server, such as the server 11, can be used in a number of environments. For example, the

16   server 11 can be used as a server maintained by a marketer or seller of a product, and can provide

17   Web pages containing images of the product. During a session during which the user may, for

18   example, wish to receive information concerning the product, the user may wish to request

19   customizations to the image, such as the orientation of the product from which the product is

20   displayed in the image, the position of the light source, the color of the product within, for example,

21   a set of colors in which the product is offered, and/or other types of customizations and be provided

22   with an image with the customizations. If the product is, for example, an automobile, types of

23   customizations may also include changes in which the automobile is displayed, including, for

24   example, the positions of one or more of the doors, illustratively, open or closed, the positions of the

25   hood or trunk. If the automobile has a sun- or moon-roof or a convertible top, customizations may

26   also include displaying the roof or top in a number of orientations. Since the server 11 has the

27   database of the scene from which the image is rendered, it can readily provide a rendered images

28   with the requested customizations without needing to provide any information from the database to

1   the user's client device 12n'. Accordingly, if the product is, for example, an automobile, the scene

2   database used by the server 11 can include information from the product design database maintained

3   by the manufacturer, and, since neither the scene database nor the information from the product

4   design database is provided to the user's client device 12n', trade secret information that may be

5   present in the information contained in the product design database will not be transferred to a device

6   that is external to the server 11. This also reduces the amount of effort required to provide the scene

7   database for the server 11, since information from the product design database can be used generally

8   directly or with few modifications in the scene database.

9   As noted above, the server 11 can be advantageously used in connection with sessions with

10 multiple client devices 12n', 12n'',..., contemporaneously, in connection with requests for images

11 generated for the same product and using the same scene database. The server 11 can selectively

12 provide that customizations requested by a user using one client system 12n' not be visible in images

13 rendered using the same scene database for a user using another client system 12n'' who has not

14 requested similar customizations. Thus, if, for example, only the user using client system 12n' has

15 requested a customization in which, with reference to the preceding example in which server 11 is

16 used to provide images of an automobile, the position of the driver's side door is changed from being

17 closed to being open by a selected amount, the server 11 will provide an image in which the driver's

18 side door is open only to the client system 12n', reflecting the customization requested only by him

19 or her, and not to the other client systems 12n'',..., even in images transmitted to the other client

20 systems 12n'',..., subsequent to the customization requested by the user of client system 12n'.

21 Similarly, if the user using client system 12n'' has requested a customization in which the color of

22 the automobile is changed from white to red, the server 11 will provide an image in which the color

23 of the automobile is red, reflecting the customization requested only by him or her, and not to the

24 other client systems 12n', 12n''',..., even in image transmitted to the other client systems 12n', 12n''',...

25 subsequent to the customization requested by the user of client system 12n''. In that case, the server

26 11 keeps track of changes to the scene database resulting from the requests from the individual client

27 devices 12n', 12n'',..., on a client-by-client basis.

1   On the other hand, as also noted above, the server 11 can selectively provide that

2   customizations requested by a user using one client system 12n' are visible in images rendered using

3   the same scene database for users using other client systems 12n", 12n"',..., concurrently engaged in

4   sessions with the server 11 in connection with the same scene database, regardless of whether the

5   latter users have requested similar customizations. The customizations may be visible in images

6   rendered for all or a subset of the other client systems 12n", 12n"',..... Thus, in that case, if a user

7   using client system 12n' has requested a customization in which, with continued reference to the

8   preceding example in which server 11 is used to provide images of an automobile, the position of

9   the driver's side door is changed from being closed to being open by a selected amount, the server

10   11 will provide an image in which the driver's side door is open to all or a selected subset of the

11   client systems 12n', 12n",..... The server 11 can provide the image with the customization to the other

12   client systems 12n", 12n"',... that are contemporaneously engaged in a session with the server 11 in

13   connection with the same scene database in response to a request therefor from each respective other

14   client system 12n", 12n"',..., which request may also include a request for a further customization.

15   Alternatively, the server 11 can, after or contemporaneous with providing the image with the

16   customization to the client system 12n' that requested the customization, also transmit the image to

17   all or a selected subset of the other client devices12n", 12n"',... that are engaged in a session

18   involving the same scene database, without a request therefor from the other respective client

19   devices. In that case, after a client device 12n', 12n", 12n"',... receives the image with the

20   customization, it can substitute the image in the Web page.

21   If the server 11 provides images with a customization, not only to the client system 12n' that

22   requested the customization, but also to all or a selected subset of the other client devices12n",

23   12n"',... that are engaged in a session involving the same scene database, without a request therefor

24   from the other respective client devices, the server 11 can also find utility in, for example, managing

25   cooperative or competitive efforts by a plurality of users using respective ones of the client devices

26   12n', 12n",..... For example, the server 11 can be advantageously used to allow a plurality of users,

27   in diverse locations using respective client devices to cooperatively design a product. In that case,

28   as the users enter customizations to a product, which can include, for example, providing an initial

-11-

1   design one or more components of the product, the server 11 can enter information defining the

2   components in the scene database. At some point, some or all of the information in the scene

3   database may be converted to a product design database, which may be used in fabricating the

4   product.

5   In addition, the server 11 can be used in connection with playing of video games over the

6   network 13. In that case, the server 11 can render images of a scene used in the came from the same

7   orientation for all of users who are playing the game, or from unique orientations for ones of the

8   users. As the users play the game over time, the server 11 can render successive images for the

9   various users and transmit them to their respective client devices 12n', 12n",... for display.

10   FIG. 2 depicts a functional block diagram of server 11 constructed in accordance with the

11   invention. With reference to FIG. 2, the server 11 includes a number of components including a

12   multiplexer module 20, a web server module 21, a script execution module 22, a user interaction

13   control module 23, a rendering control module 24, a script store 25, a model store 26 and a rendering

14   engine 27. The multiplexer module 20 connects to the network 13 and receives information retrieval

15   requests from a user's client device 12n. Generally, a user using client device 12n will provide input

16   entered through, for example, a user input device to input request information to a browser 14, and

17   the browser will generate one or more information retrieval requests for transmission to the server

18   11 requesting retrieval of a Web page. The multiplexer 20, in turn, receives information retrieval

19   requests, which may be a Web page retrieval request or, as will be described below, an image

20   retrieval request, from the network 13. When the multiplexer 20 receives an information retrieval

21   request from the network 13, it will either respond to the Web page retrieval request itself, or it will

22   transfer the request to one of the web server 21 or the user interaction control module 23. If the

23   information retrieval request is the first request from the browser for a session, the multiplexer 20

24   will generate a response that includes a user identification (UID) for the session, and transmit the

25   response to the client device 12n for use by the browser 14.   Subsequent information retrieval

26   requests generated by the browser 14 for transmission to the image rendering device 11 for the

27   session will include the UID, and the image rendering device 11 will use the UID to identify the

1   session and keep track of the particular user for which images and customizations have been

2   requested during the session.

3   After the browser 14 receives the response, including the UID, generated by the multiplexer

4   20 and transmitted thereby over the network 13, the browser 14 will generate a new Web page

5   retrieval request for transmission by the client device 12n to the server 11. The new Web page

6   retrieval request will generally correspond to the previous Web page retrieval request, except that

7   it will also include the UID received in the response that had previously been received from the

8   multiplexer 20. When the multiplexer 20 receives a Web page retrieval request from the client

9   device 12n that includes a UID, it will provide the Web page retrieval request to the web server 21.

10   The web server 21, in turn, will provide information from the request to the script execution module

11   22, which, using one or more scripts from a script store 25 and information provided by the user

12   interaction control module 23, will generate at least a portion of a Web page for transmission by the

13   multiplexer 20 to the client 12n.

14   The portion of the Web page that is generated by the web server 21 will include at least the

15   textual portion of the Web page requested in the Web page retrieval request, and in one embodiment

16   will be generated in the well-known HyperText Markup Language (HTML). The Web page that is

17   generated may include links identifying, for example, one or more images that are to be displayed

18   as part of the Web page. The links are augmented to identify the UID that was included with the

19   Web page retrieval request. After the web server 21 and script execution module 22 have generated

20   the portion of the Web page to be provided thereby, they can provide the Web page to the

21   multiplexer 20 for transmission over the network 13 to the client device 12n.

22   The script execution module 22 will also provide information to the user interaction control

23   module 23, including the UID that was received in the Web page retrieval request, as well as

24   identification of any customizations to one or more images that are to be displayed as part of the

25   Web page that were requested in the Web page retrieval request. Since the Web page retrieval

26   request is the first retrieval request for the particular Web page for the session, the Web page

27   retrieval request generally will not include any customizations. The user interaction control module

1   23 can perform some preliminary processing operations to prepare to render the image(s) when it

2   receives a request therefor from the multiplexer 20, as will be described below.

3   When the client device 12n receives the portion of the Web page as generated by the web

4   server 21 and script execution module 22 from the server, it can provide it to the browser 14. The

5   browser 14 can display the received portion, and use the links, as augmented to identify the UID for

6   the session, to generate one or more requests, which will typically be image retrieval requests, that

7   are associated with respective ones of the links, to initiate retrieval of respective images for display

8   as part of the Web page. Image retrieval request(s) generated by the browser 14 for the respective

9   images will include both the image identification information from the respective links, as well as

10   the UID with which the links were augmented, to identify the session for which the server 11 is to

11   render the images. The client device 12n transmits the image retrieval requests to the server 11.

12   The image retrieval requests will be received by the multiplexer 20 and forwarded directly

13   to the user interaction control module 23 for processing, bypassing the web server 21 and script

14   execution module 22. The user interaction control module 23, rendering control module 24 and the

15   rendering engine 27, will render the respective images and provide them to the multiplexer 20 for

16   forwarding over the network 13 to the client device 12n. Generally, the rendering control module

17   24 will control the rendering operations in connection with information in a scene database that it

18   maintains. The scene database contains information useful in connection with rendering of an

19   image, including

20   (i) a three-dimensional representation of at least a portion of a scene from which a two-

21   dimensional image is to be generated,

22   (ii) information as to the positions and orientations of light source(s) that are used to

23   illuminate the object(s) in the scene, and

24   (iii) the position(s) and orientation(s) of camera(s) that are to be simulated in rendering of

25   the image, relative to the object(s) in the scene, as well as information as to the optical characteristics

26   of the camera(s), such as, for example, the camera(s) magnification or zoom settings.

1   If Web page retrieval request received requesting an image of a scene did not include any requests

2   for customizations, as will be generally the case for the first image retrieval request for a particular

3   session, the user interaction control module 23 will enable the rendering control module 24 to render

4   the image with selected characteristics, including, for example, particular portions of the scene, light

5   source(s) in particular position(s) relative to the scene, a respective camera in a particular position,

6   orientation and with particular optical characteristics, and the like. The characteristics can be default

7   characteristics, the last valid characteristics of the scene as may be stored in and retrieved from a user

8   database, or other characteristics as will be apparent to those skilled in the art. The image(s) can be

9   provided in any convenient form including, for example, as a bitmap or compressed using any of the

10   well-known compression methodologies. The client device 12n will provide the image(s) to the

11   browser 14, which can display them in regions of the displayed web page reserved therefor.

12   The Web page displayed by the browser 14 may provide tools or other controls that would

13   allow the user to request customizations of the scene represented by the image. After the user has

14   manipulated one or more of the tools to request one or more customizations, the browser 14 can

15   generate a Web page retrieval request for the same Web page, but with customization information

16   specified for at least one of the images rendered with the scene. The Web page retrieval request will

17   identify, for each image for which customization information is provided, the particular image, as

18   well as the customizations that are to be performed in connection with the scene from which the

19   particular image is to be rendered. A number of types of customizations may be specified, including,

20   for example

21   (i) translation and/or rotation of one or more of the objects in the scene relative to each other

22   or to a coordinate system;

23   (ii) addition of objects to, or deletion of objects in, the scene;

24   (iii) changes to the forms of the objects in the scene;

25   (iv) changes of the material characteristics of objects in the scene

26   (v) the merging of a plurality of scenes to form a new scene;

-15-

1   (vi) changes to the positions and orientations of light source(s) that are used to illuminate the

2   object(s) in the scene;

3   (vii) changes to the position(s) and orientation(s) of camera(s) that are to be simulated in

4   rendering of the image, relative to the object(s) in the scene, as well as changes to the optical

5   characteristics of the camera(s), such as, for example, the camera(s) magnification or zoom settings,

6   (viii) high-level compound commands, such as "begin driving this car through a city,"

7   and other types of customizations as will be apparent to those skilled in the art. In addition, the Web

8   page retrieval request will include the UID provided by the multiplexer 20 at the beginning of the

9   session. After the browser 14 has generated the Web page retrieval request, the client device 12n

10   will transmit the Web page retrieval request to the server 11 over the network 13.

11   As before, the multiplexer 20 will receive the Web page retrieval request and provide it to

12   the web server 21. The web server 21 and script execution module 22 will generate the HTML

13   portion of the requested Web page and provide it to the multiplexer 20 for transmission to the client

14   12n, which will include links to the image or images that are to be displayed on the Web page,

15   which, as before, have been augmented with the UID identifying the session for the user. In

16   addition, the web server 21 and script execution module 22 will provide information to the user

17   interaction control module 23 as to

18   (i) the scene(s) for which customization(s) that have been requested,

19   (ii) the particular customizations that were requested to the respective scenes, and

20   (iii) the UID identifying the session for which the customizations have been requested. It

21   will be appreciated that the user interaction control module 23 can use the UID identifying the

22   session for which the customizations have been requested to associate the particular session for

23   which the customizations have been requested. After the user interaction control module 23 receives

24   the customizations that have been requested and the session for which the customizations have been

1   requested, it can perform some preliminary processing operations to prepare to render the image(s)

2   with the respective customizations, as will be described below.

3   After the client device 12n receives the HTML portion of the Web page from the server 11,

4   it will provide it to the browser 14, which can display that portion of the Web page in the same

5   manner as before. In addition, as before, the browser 14 will generate one or more image retrieval

6   requests to initiate retrieval of the image(s) from the server 11. Further, as before, the image

7   retrieval requests will include information from the link(s) that identify the particular image(s) that

8   are to be retrieved, as well as the UID of the session as provided in the augmented links. The image

9   retrieval request(s) will be provided to the client device 12n, which, in turn, will transmit the image

10   retrieval request(s) to the server 11.

11   As before, the image retrieval request(s) will be received by the multiplexer 20, which will

12   provide them to the user interaction control module 23. The user interaction control module 23 will

13   enable the rendering control module 24 to render the image(s) and provide the rendered image(s) to

14   the multiplexer 20 for transmission to the client device 12n. In rendering each image, the user

15   interaction control module 23 will enable the rendering control module 24 to render the respective

16   image with the customization(s), if any, that the user requested in the Web page retrieval request.

17   In that operation, if a customization to an image is such as would require customization of the scene

18   as stored in the scene database, the user interaction control module 23 will provide appropriate

19   customizations to the scene database. Depending on the environment in which the server 11 is used,

20   the customization(s) may be such as to provide that they will be used only in connection with the

21   image rendered for the session associated with the particular UID, or, alternatively, in connection

22   with images rendered for all or a subset of the sessions that are contemporaneously engaged in

23   sessions with the same scene in the scene database. In addition, if a customization provides a

24   change to the scene database, including, for example, viewing orientation or the position of the light

25   source(s) illuminating the scene, addition of objects to or deletion of objects from the scene, the user

26   interaction control module 23 can provide information thereof to the rendering control module 24

27   for use in rendering. If a customization requires an addition of an object to the scene, the user

28   interaction control module 23 can, for example, enable the rendering control module 24 retrieve

-17-

1   information from the model database 26 describing the object to be added to the scene. Thereafter,

2   the user interaction control module 23 can enable the rendering control module 24 to render an

3   image, which the rendering control module 24 will provide to the user interaction control module

4   23. The user interaction control module 23 will, in turn, provide the image to the multiplexer 20,

5   for transmission to the client device 12, for display by the browser. These operations can be

6   performed for each of the images for which image retrieval requests have been received.

7   With this background, the structure and operation of the user interaction control module 23

8   and the rendering control module will be described in more detail in connection with FIG. 2. With

9   continued reference to FIG. 2, the user interaction control module 23 includes several components,

10 including a user manager 30, a connection manager 31, an event manager 32, a model manager 33

11 and a plurality of operators. Generally, in one embodiment, the user interaction control module 23

12 makes use of operators to perform operations. One operator is a socket gateway operator 34, which

13 receives UID and customization information from the script execution module and image retrieval

14 requests from the multiplexer 20, and provides rendered images to the multiplexer 20 for transfer

15 to respective ones of clients 12n. To facilitate rendering of an image, the operators can also be

16 linked together into a graph 35, with the particular operators and sequence thereof in the operator

17 graph being selected to facilitate generation of the image having the desired characteristics. If, for

18 example, an image is to be generated using a scene, and the user has provided a Web page retrieval

19 request in which no customizations have been requested to the scene, an image of which is to be

20 rendered for display on the Web page, which may be the case if, for example, the Web page retrieval

21 request was the first request for the Web page during the respective session, the operator graph may

22 be a default graph for use in rendering images using the particular model.

23 It will be appreciated that the particular set of operators that are provided for the user

24 interaction control module 23 will depend on the particular environment in which the server 11 is

25 used. In addition to a socket operator for use as the socket gateway, operators that may be

26 advantageously used in connection with the user interaction control module 23 may include such

27 operators as a operators of the object translation operator type, operators of an object rotation

28 operator type, operators of a color operator type, operators of a timekeeper operator type, and

-18-

1   operators of a render operator type. An operator of the translation operator type can be used to

2   facilitate updating of the model for a scene in the scene database to translate an object in the scene

3   by a selected distance in a selected direction. For example, if the server 11 is to be used in

4   connection with a game in which objects in the scene are to be moved along a path, an operator of

5   the object translation operator type can be used to translate an object in the scene along a path in the

6   scene. The particular object that is moved, as well as the distance and direction that the object is

7   moved, can be specified as parameters whose values are determined by the image customization

8   information. It will be appreciated that the extent to which an object can actually be moved may be

9   constrained by other features of the scene, including, for example other objects that may be present

10   in the scene.

11   An operator of the rotation operator type can be used to facilitate updating of the scene in

12   the scene database to rotate an object in the scene by a selected angle around a selected axis. The

13   particular object that is rotated, as well as the angle and direction that the object is rotated, as well

14   as possibly the axis of rotation, can be specified as parameters whose values are determined by the

15   image customization information. For example, if the particular object that is to be rotated is an

16   automobile door, the axis of rotation will comprise the axis specified by the door's hinges, which

17   may be determined by the model of the automobile as stored in the scene database. The angle that

18   the door is rotated around the axis, and the direction of rotation can be specified as parameters whose

19   values are determined by the image customization. It will be appreciated that the angle that the

20   object may be rotated around any particular direction may be constrained by other objects in the

21   scene, including other components of the automobile.

22   An operator of the color operator type can be used to change the color of at least a portion

23   of the surface of an object in the scene. A color operator can operate by editing parameters of

24   shaders that are provided in the scene database in response to image customization information.

25   An operator of the timekeeping operator type can be used to provide a time stamp or value

26   to respective client devices 12n. If the server 11 is being used in connection with, for example, a

27   game, the time stamp or value as generated by the timekeeping operator can be transmitted to the

1   client devices 12n of all of the users who are playing the game to provide them with a common time

2   reference. The time stamp or value can identify the particular time as determined by the server 11,

3   and can be provided to, for example, the script execution module 22 for use in the HTML portion

4   of the Web page when it generates that portion for transmission to a client device 12n.

5   An operator of the rendering operator type can be used to initiate and control rendering of

6   an image by the rendering engine 27.

7   Other types of operators useful in the server 11 will be appreciated by those skilled in the art.

8   For example, an operator type can be provided to control the position, angular orientation,

9   zoom/focal length, aperture setting, and so forth of a camera. In addition, an operator type can be

10   provided to control the position, angular orientation, color, brightness, and so forth of a light source.

11   In addition, a high-level operator type can build on and utilize operators of other operator types to

12   perform compound operations such as moving an object on a motion path controlled by gravity or

13   moving a car door between the "open" and "closed" states automatically.

14   At least some of the types of operators may also be of one of two subtypes, including a

15   private subtype and a public subtype. An operator of the private subtype is used to provide a

16   customization that is only visible in the image(s) that are to be subsequently rendered of the

17   particular scene for the particular session identified by the particular UID that is associated with the

18   image customization information that requested the customization, or for a selected subset of UID's

19   that are contemporaneously using the scene. On the other hand, an operator of the public subtype

20   is used to provide a customization that is visible in the image(s) that are to be subsequently rendered

21   of the particular scene for all or possibly a larger subset of UID's that are contemporaneously using

22   the scene. The server 11 provides for several privatization levels, so that, for a lower level, a

23   customization will be visible in images that are subsequently rendered of the particular scene for the

24   particular session identified by the selected subset of UID's that are contemporaneously using the

25   scene, and, for a higher level, a customization will be visible only in the image(s) that are to be

26   subsequently rendered of the particular scene for the particular session identified by the particular

27   UID that is associated with the image customization information that requested the customization.

1   The user interaction control module 23, in generating an operator graph 25 to facilitate
2   rendering of an image of a scene, will instantiate operators of the required types and link them
3   together to form the operator graph 35. The user interaction control module 23 can perform these
4   operations after it receives the image customization information and UID associated with a Web
5   page retrieval request from the script execution module. This will allow the operator graph to be
6   ready when an image retrieval request is received for the particular image. After the image retrieval
7   request, the user interaction control module 23 can initiate execution of the various operators in the
8   graph to, in turn, initiate rendering of the image.

9   The user manager 30 is provided to keep track of UID's and converts between UID's and
10   identifiers, referred to as RID's (rendering session identifiers) that are used to keep track of rendering
11   sessions by the rendering control module 24.

12   The model manager 33 and connection manager 31 cooperate to create operator graphs 35
13   in response to image customization information provided by the script execution module 22 in
14   response to Web page retrieval requests from the respective client devices 12n. When the model
15   manager 33 receives image customization information from the script execution module 22 relating
16   to an image that is to be rendered, it will determine the operators of the respective types that are to
17   be used in the operator graph 35, instantiate the operators and determine the topology of the operator
18   graph, that is, how the operators are to be linked together to form an operator graph 35. The
19   connection manager 31 will perform the actual linking. Each operator has at least one input and an
20   output, and the model manager 33 will determine, for each input of each operator, the respective
21   operator that is to provide a value or status information for the respective input. The operator that
22   is to provide a value or status information for an input of an operator is upstream of that operator,
23   and the operator whose input is to receive the value or status information is downstream of the
24   operator that is to provide the value or status information. The connection manager 31 will perform
25   the actual linking of the operators that have been instantiated by the model manager to form the
26   operator graph 35.

1   The model manager 33 can determine the types of operators that are to be used from the

2   image customization information that is provided by the script execution module 22. For example,

3   if the image customization information is to enable an object to be translated a predetermined

4   distance in a particular direction, the model manager 33 can instantiate an operator of the translation

5   operator type, and provide as parameters such information as, for example, the identification of the

6   object that is to be translated, the direction that the object is to be translated and the displacement

7   along the particular direction. Similarly, if the object is to be rotated around an axis, the model

8   manager 33 can instantiate an operator of the rotation operator type, and provide as parameters such

9   information as the identification of the object that is to be rotated, the position and orientation of the

10   axis around which the object is to be rotated, the direction around the axis that the rotation is to take

11   place and the angle through which the object is to be rotated. In addition, if the color of the object

12   is to be customized from a default color, the model manager 33 can instantiate an operator of the

13   color operator type, and provide as a parameter the color to which the object is to be customized.

14   The model manager 33 and connection manager 31 can instantiate and link corresponding

15   operators for every object that is to be translated, rotated and whose color is to be customized, and

16   link them into the operator graph that is to be used to control the rendering of the image associated

17   with the image customization information. The particular order in which the operators are connected

18   in an operator graph 35 may be determined by several factors, including whether the operators

19   commute, that is, whether, if image customization information requires usage of operators of two

20   types, the two operators can be applied in any order and provide the same result. Generally, for

21   example, if an image customization requires operators of the translation operator, rotation operator

22   and color operator type, operators of the respective type can be applied in any order. Generally, the

23   operator of the rendering operator type will be expected to be one of the last operators in the operator

24   graph 35, after all of the operators instantiated to update the model of the scene in the model

25   database 26. The socket gateway 34, which, as noted above, is also an operator, will be both the first

26   and the last operator in the operator graph 35. When the multiplexer 20 receives an image retrieval

27   request from a client 13n, the multiplexer 20 will notify the socket gateway and provide information

28   identifying the UID and an identifier identifying the image that is to be rendered to the socket

1   gateway 34 to initiate execution of the operator graph to facilitate updating of the model of the scene

2   in the model database 26, if necessary, and rendering of the image. After the image is rendered, it

3   will be provided to the socket gateway 34 for provision to the multiplexer 20 and transmission to the

4   particular client device 12n that issued the image retrieval request.

5   The event manager 33 controls execution of the operators that comprise an operator graph

6   35 and, in that case, manages events that occur during execution. Generally, the event manager 33

7   controls execution of the operator graph 35 according to a "data flow" paradigm, in which an

8   operator in an operator graph 35 is executed when all of its inputs, which include both values of

9   parameters provided by the image customization information provided in the Web page retrieval

10   request and values and/or status information that are provided by operators that are upstream of the

11   respective operator in the operator graph 35, have been provided with respective values and/or status

12   information. The status information may merely indicate that an upstream operator in the graph has

13   finished execution. Accordingly, if, for example an operator graph 35 includes operators of the

14   translation operator, rotation operator and color operator type, to translate, rotate and change the

15   color of the same object, followed by an operator of the rendering operator type, the event manager

16   may enable the operators of the translation operator, rotation operator and color operator type in any

17   order. Before the operator of the rendering operator type can be executed, the operators of the

18   translation operator, rotation operator and color operator types will need to provide status

19   information that indicates that they have successfully finished execution. As each respective

20   operator of the translation operator, rotation operator and color operator type is executed, it will

21   update the scene in the scene database, and will generate status information to indicate when they

22   are finished, which is provided to the rendering operator as an input. After all of the operators of the

23   translation operator, rotation operator and color operator type, as well as operators of other types that

24   may be provided, have finished execution, the event manager 33 will note that all of the inputs to

25   the rendering operator have received status information indicating that the operators upstream thereto

26   had successfully completed, the event manager 32 can enable the operator of the rendering operator

27   type to be executed. The event manager 32 can enable operator graphs comprising instantiated

1   operators of any combination of operator types, connected in any of a number of topologies, to be

2   executed in a similar manner.

3   The event manager 32 can initiate execution of an operator graph 35 at any operator in the

4   operator graph 35. As will be described below in more detail, if during execution of an operator, an

5   input value is needed from another operator to allow for continued execution of the operator,

6   execution of the one operator can be suspended, and the other operator executed to allow for

7   generation of the input value that is needed by the one operator. By allowing for execution of

8   operators in this manner, if it turns out that a value generated during execution of one operator is not

9   needed for another operator, which may occur if, for example, rendering of an image is aborted.

10   As noted above, the model manager 33, in response to image customization information from

11   the script execution module 22, initiates creation of an operator graph 35 to facilitate rendering of

12   an image of a scene with the customizations, if any, requested in a Web page retrieval request. In

13   addition, the model manager 33 can enable models of objects, that are stored in the model database

14   26, and that may be needed in the scene database associated with a scene, to be loaded into the

15   respective scene database.

16   The rendering control module 24, while an operator graph 35 is being executed to facilitate

17   rendering of an image, controls updating of a scene from which the image is to be rendered as

18   represented in the scene database in response to execution of operators of the respective types. In

19   addition, the rendering control module 24 controls rendering of the image during execution of an

20   respective operator of the rendering operator type. The rendering control module 24 comprises an

21   API control module 40, a job manager 41, a world/session/transaction manager 42, and one or more

22   scene databases 43. The rendering engine 27 performs actual tessellating and rendering operations

23   during execution of an operators of the respective operator type.

24   The world/session/transaction manager 42 manages "worlds," "sessions" and "transactions."

25   Generally, a transaction bounds a set of consistent database operations in connection with the scene

26   database 43. For example, rendering is considered a transaction because rendering requires a

27   consistent view of the scene in the scene database 43 that must not be changed during the rendering

1   operation. Similarly, a modification to a scene is also a transaction, since typically a modification

2   to the scene requires incremental changes to many scene data elements in the scene database all of

3   which need to be performed as a unit to ensure that the scene in the scene database 43 remains

4   consistent. For example, generally, an object that is actually present in the scene is represented by

5   an object type and an instance, so that, if a scene contains two objects of the same object type, that

6   can be represented in the scene database 43 by one object type and two instances, all of which are

7   scene data elements. If it is necessary to delete an object, represented by object type and instance

8   scene data elements, both the object type and instance scene data elements will need to be deleted

9   as a unit. This will ensure that problems do not arise in connection with the rendering engine 27,

10   which can occur if, for example, during rendering of the scene, the object type scene data element

11   has been deleted but not the instance scene data element, since at some point the rendering engine

12   27 will need to attempt to access the deleted object type scene data element.

13   A session as managed by the world/session/transaction manager 42 generally corresponds

14   to a session with a client device 12n. Worlds are used to disambiguate multiple scenes that may be

15   used for client devices 12n that are concurrently engaged in sessions with the server 11, and

16   particularly may be used to disambiguate scene data elements of the respective scenes that may have

17   similar names. For example, if two different scenes both call their cameras "cam," if a customization

18   to the camera is requested for a scene being used for one session, that customization should only be

19   made in that scene, and not in the other scene.

20   The API control module 40 controls updating of the scene and rendering of an image in one

21   or more jobs, and the job manager 41 schedules the jobs based on selected criteria. In one

22   embodiment, the criteria includes such issues as job age, whether a job is a prerequisite for a number

23   of other jobs, job "cost," and other criteria as will be described in more detail below. The age of a

24   job can be a desirable criteria since delaying processing of a job based on other criteria can be

25   undesirable to completion of rendering of the image(s) that depend on the delayed job. The job

26   "cost" criteria may be a function of other criteria including, for example, an estimate of the

27   processing time required to execute the job, or to finish execution of the job if execution is

28   suspended, an estimate of the amount of various processing resources, such as memory, that may be

1   required, and the like. A job for which the estimate of the processing time and/or required

2   processing resources is higher will generally have a higher cost associated therewith than a job for

3   which the estimate is lower. In one embodiment, jobs associated with higher costs will be processed

4   on a preferential basis over jobs with lower costs, which can increase the likelihood of that the jobs

5   relating to rendering of an image, which may be processed in parallel, can be completed in less time

6   than otherwise.

7   As noted above, the rendering engine 27 performs the actual rendering operations required

8   to render the image. In one embodiment, the rendering engine 27 comprises Mental Ray Version

9   3.0, available from Mental Images, G.m.b.H., & Co., KG, Berlin, Germany, although other rendering

10   engines, such as OpenGL can be used. Generally, the rendering engine 27 performs tessellation of

11   the scene data elements of a scene in the scene database 43 as necessary prior to rendering an image

12   of the scene, and thereafter renders an image. The rendering engine 27 need not tessellate all of the

13   scene data elements of a scene before it begins rendering an image; the rendering image 27 can

14   instead tessellate a portion of the scene data elements before rendering an image of the portion that

15   has been tessellated, and repeat these operations as necessary to render the image as desired. It will

16   be appreciated that the rendering engine 27 need only tessellate portions of those scene data elements

17   that will be depicted in an image that is to be rendered.

18   The elements of the rendering control module 24 will be described in detail in connection

19   with FIG. 2. As noted above, the rendering control module 24 comprises the API control module

20   40, the job manager 41, the world/session/transaction manager 42 and the scene database 43. The

21   API control module 40 operates as the interface among the user interaction control module 23 and

22   the other elements of the rendering control module 24 and the rendering engine 27. In that function,

23   the API control module 40 receives calls from the user interaction control module 23 when an

24   operator graph 35 is executed, and provides status information and, while an image is being rendered

25   or after the image has been rendered, provide either the portions rendered or the entire image to the

26   user interaction control module 23. The status information provided by the API control module 40

27   may be used by elements of the user interaction control module 23 during execution of an operator

28   graph 35. For example, as noted above, operators comprising an operator graph 35 may be executed

1   in any order and, if, in executing an operator, the rendering control module 24 determines that it

2   needs an input value from an operator that is upstream of the operator being executed, the API

3   control module 40 will provide status information to so notify the user interaction control module

4   23. After receiving the status information, the user interaction control module 23, in particular the

5   event manager 32, can enable the operator that is to provide the needed value to be executed. Any

6   number of such sequences, one per user, may be in progress simultaneously

7   Execution of an operator generally entails performing one or more jobs. For example,

8   loading an object from the model store 26 into the scene database 43 can entail several jobs,

9   including, for example, retrieval of data describing the object from the model store, converting the

10   data from, for example, a form that might be used by a computer-assisted design ("CAD") system

11   to a form that would be useful to the rendering engine 27, and loading the converted data into the

12   scene database 43 and linking it to the respective scene. Each of these operations can be performed

13   as a respective job, or multiple jobs. Similarly, rendering an image can entail several jobs, including

14   rendering a rectangle, loading a texture, tessellating a surface, and so forth. The job manager 41

15   manages the jobs that are concurrently being executed so they will be executed in an efficient

16   manner. This will allow the server 11 to provide Web pages and rendered images to a number of

17   clients 12n concurrently with minimal delay. The job manager 41 maintains a dependency graph

18   44 of jobs that are to be executed, with each of the jobs being annotated with a job cost value

19   indicating the respective job's job cost value as described above. When a job of a particular type is

20   first linked into the dependency graph 44 of jobs to be executed, it can be accompanied by a job cost

21   value that is an initial estimate. As the job manager 41 executes jobs of the respective job type, it

22   can keep track of the resources that are used and update the job cost value for use when a job of the

23   same type is subsequently executed.

24   As noted above, operators can be public or private, with several levels of privacy. Scene

25   elements in the scene database 43 can also be public or private, with corresponding levels of privacy.

26   If, for example, a private translation operator is provided in an operator graph associated with a

27   particular UID to facilitate translation of an object in a scene, the scene element(s) in the scene

28   database 43 that represents the object as translated will be also be private. In addition, jobs are of

-27-

1   public and private subtypes, with corresponding levels of privacy, and the job(s) that are executed

2   during execution of an operator of the public will correspond to the public/private subtype of the

3   operator for which they are executed.

4   The job manager 41 enables other modules, such as the world/session/transaction manager

5   42 and, in particular, the rendering engine 27, to perform the individual jobs, and in addition controls

6   access to the scene information in the scene database 43. In one embodiment, the scene database 43

7   is in the form of a cache, in which design information for objects that are to in a scene can be loaded

8   using model information from the model store 26. As the amount of data in the scene database 43

9   increases, the job manager 41 can select data to be removed from the scene database 43. A number

10   of selection criteria can be used to determine which data is to be removed from the scene database

11   43. One embodiment makes use of a pin counter (not shown) associated with each element of screen

12   data. When a module that needs to make use of an element of scene data issues an access request

13   for the element, it increments the pin counter, and when the module is finished with the element, it

14   decrements the pin counter. Each element of scene data is also associated with an access sequence

15   value, and, when a module issues an access request for an element, an integer is incremented and

16   provided as the element's access sequence value. Each scene data element whose associated pin

17   counter has the value "zero" is a scene data element for which all of the modules that needed to use

18   the scene data element concurrently have finished using the scene data element. In that case, the

19   scene data element can be deleted from the scene database 43. It will be appreciated, however, that,

20   although no modules are then using the scene data element at that point in time, a module may

21   subsequently need to use the scene data element.

22   Since generally the likelihood that a module will subsequently need to use a scene data

23   element will decrease the longer it has been since it was last used, the job manager 41 can sort the

24   scene data elements whose pin counters have the value "zero" by access sequence values; it will be

25   appreciated that the least recently accessed scene data elements will be those for which access

26   sequence values are relatively low and the most recently accessed scene data elements will be those

27   for which access sequence values are relatively high. Preferably, the job manager 41 will select one

28   or more scene data elements for deletion for which access sequence values are relatively low on the

1   sorted list. However, since it will generally not be efficient to simply delete the oldest scene data

2   element, since that scene data element may be relatively small, the job manager 41 also takes the size

3   of the respective scene data elements into account when selecting scene data elements for deletion,

4   by scaling the scene data elements' sizes in relation to their relative position on the sorted list, and

5   selecting for deletion the scene data elements in order of their scaled sizes. That is, if there are "n"

6   scene data elements in the sorted list, i=0, 1, 2,...,n-1, where i=0 is the last scene data element (that

7   is, most recently accessed, the scene data element having the highest sequence value) on the list, and

8   i=n-1 is the first scene data element (that is, least recently accessed, the scene data element having

9   the lowest sequence value) on the list, the job manager 41 will generate scaled size values according

10   to $size_{scaled}^{i} = \dfrac{size_i}{n-i}$. It will be appreciated that the scaled size value $size_{scaled}^{n-1}$ for the first (that

11   is, the oldest) scene data element i=n-1 will correspond to its size $size_{n-1}$, the scaled size value

12     $size_{scaled}^{n-2}$ of the second scene data element i=n-2, will correspond to one-half its size $size_{n-2}$, and so

13   forth, and the scaled size value $size_{scaled}^{0}$ of the last scene data element i=0 will correspond to $\dfrac{1}{n}$

14   its size $size_0$. After generating the scaled size values, the job manager 41 will select one or more

15   scene data elements, from the largest to the smallest according to their scaled size values $size_{scaled}^{i}$

16   for deletion, as may be necessary to accommodate scene data elements that are to be loaded into the

17   scene database 43.

18   The job manager 41 creates a job for linkage into the dependency graph 44 in response to a

19   request therefor during execution of operators in an operator graph 35, or from a module, such as

20   rendering engine 27 or world/session/transaction manager 42, while those modules are executing

21   other jobs. When an operator in an operator graph 35 requests creation of a job, the request is

22   provided to the job manager 41 through the API control module 40. Similarly, when the rendering

23   engine 27 or world/session/transaction manager 42 requests creation of a job, the request is provided

24   to the job manager 41 through the API control module 40. Generally, a job is a request to create a

1   result data set based on values of one or more parameters. The result data set will be stored in the

2   scene database 43. The parameters describe the operation to be performed during execution of the

3   job and possibly other scene data elements that may be present in the scene database 43. In response

4   to a job creation request, the job manager 41 generates a job description data structure that includes

5   such information as:

6   (i) an operation code that identifies the type of operation to be performed during execution

7   of the job;

8   (ii) a module identifier identifying the particular module that is to execute the job;

9   (iii) a status identifier that identifies the status of the job; in one embodiment, possible status

10   identifiers include pending, running, suspended, finished, done, flushed and failed;

11   (iv) one or more identifiers that identify the scene data element(s) in the scene database that

12   are to be used in executing the job;

13   (v) one or more identifiers that identify storage locations in the scene database in which

14   results are to be stored during execution of the job;

15   (vi) an estimate of the job cost value;

16   (vii) an actual job cost value if the job has finished executing and the status is "done";

17   (viii) the identification of any prerequisite jobs, that is, jobs that need to be executed before

18   this job begins execution; and

19   (ix) control information such as the identification of the particular thread, on a processing

20   platform that executes programs in threads, or similar structure (generally, "thread"), that is expected

21   to execute the job if the status is pending, or that is executing the job if the job is being executed,

22   caching control information, and so forth.

1   Items (i), (iv), and (v) can generally be provided as parameters in the job creation request.

2   Generally, the job cost value estimate (item (vi) above) can be based on a number of criteria,

3   including, for example,

4   (a) the apparent complexity of the job, which may be based on, for example, the number of

5   vertices to tessellate if the job is one to tessellate, the number of pixels to render if the job is one to

6   render, the number of photons to cast if the job is one to cast photons from a light source, and the

7   like, which may be adjusted by options such as sampling densities;

8   (b) the estimated memory requirements for the job; and

9   (c) data transfer delays that may be incurred in obtaining data to be used in executing the job

10   if the prerequisite jobs are executed by different host computers.

11   In addition, additional factors can be used in connection with a job cost value estimate, including

12   factors such as the number of jobs of the same type that are suspended on the thread that is expected

13   to execute the job or that is executing the job, and other factors as will be apparent to those skilled

14   in the art. It will be appreciated that the job manager 41 may update a job's job cost value estimate

15   in view of changes in, for example, the additional factors noted above.

16   The job manager 41 uses a number of criteria in selecting a job for execution, including the

17   number of known unresolved prerequisites, that is, jobs that need to be executed before the

18   respective job, to provide a value therefor, but that have not yet been executed, as well as the jobs'

19   job cost value estimate. Preferably, the job manager 41 will select a job with no or relatively few

20   unresolved prerequisites. On the other hand, if a job is a direct or indirect prerequisite for a number

21   of other jobs, it will preferably be executed before jobs that are prerequisites for fewer jobs. In

22   addition, preferably, the job manager will preferably select a job with a relatively high job cost value

23   estimate for execution, which can improve finalization parallelism, as long as respective threads do

24   not exceed a particular concurrent aggregate job cost values for jobs assigned thereto. Generally,

25   the job manager 41 will assign jobs to threads using a number of criteria, including maximizing data

26   locality, minimizing and balancing the number of suspended jobs on a thread to keep stacks small,

1   and other criteria as will be apparent to those skilled in the art. In addition, if, during execution of

2   a job, it is discovered that another job, which was not previously known to be a prerequisite, is in

3   fact a prerequisite, the job manager 41 will preferably assign the newly-discovered prerequisite to

4   the same thread as the job for which it is a prerequisite, for similar reasons.

5   Generally, the job manager 41 executes jobs as they are necessary to provide data for other

6   jobs. For example, texture image elements are created as placeholders in the scene database 43, but

7   they are empty and have no associated pixel array. When they are accessed for the first time, filing

8   the texture image element necessitates executing an associated texture load job. A job that is

9   accessing the texture image element will be suspended until a texture load job is executed. As

10   another example, generally, an object that has not yet been tessellated may initially be coarsely

11   represented in the scene database 43 by a placeholder in the form of a bounding box. A bounding

12   box may be in the form of, for example, a geometric object having surfaces in the form of, for

13   example, triangles, quadrilaterals or the like, that coarsely bounds the actual object associated

14   therewith. When a ray is sent into a scene, the ray may hit the bounding box. A placeholder for the

15   bounding box will already exist in the scene database 43, but the placeholder will not contain the

16   scene data for the actual object represented by the bounding box. Instead, the placeholder contains

17   a pointer to the job description for the job that is provided to generate the scene data. When the

18   object represented by the bounding box is illuminated, a ray generated to represent the illumination

19   would hit one of the triangles, quadrilaterals, and so forth, that comprises the bounding box in the

20   scene. When that occurs, the scene data representing at least a portion of the actual object will be

21   need to be retrieved from the model store 26, tessellated and stored in the scene database 43, and the

22   job manager 41 will enable respective jobs therefor to be executed, and a job that might be accessing

23   the object will need to be suspended until those jobs are executed. The bounding box essentially

24   operates as a placeholder for the object. The bounding box will not be deleted from the scene

25   database 43 when the object is stored therein, since, as noted above, the object might be deleted from

26   the scene database 43.

27   More generally, when a job being executed makes an access to a scene data element in the

28   scene database 43, the following operations are performed:

-32-

1   (i) a module initiates an access to a scene data element in the scene database 43;

2   (ii) if the scene database 43 that contains

3   (a)   a valid scene data element, that is, a scene data element that is not a placeholder, it

4   will provide the scene data element to the module that initiated the access, after

5   which the module can make use of the scene data element; but

6   (b)   an invalid scene data element, that is, a scene data element that is a placeholder for

7   data that has not yet been created but that is associated with a job description, it will

8   provide a notification thereof to the job manager 41;

9   (iii) the job manager 41 will update its job dependency graph 44 to reflect the job for which

10   it received notification from the scene database 43, thereby to enable the job to be executed at some

11   point in the future;

12   (iv) the job manager 41 also suspends execution of the thread containing the module that

13   initiated the scene data element access;

14   (v) the job manager 41 selects a new job and assigns it to the thread whose execution was

15   suspended; the selection of the job is based on the estimated job cost values of the jobs in the

16   dependency graph 44, and, accordingly, the selected job may, but need not, be the job that was added

17   to the dependency graph 44 (reference item (iii) above); and

18   (vi) when a job is finished executing, it will enable data to be stored in the placeholder for

19   the scene data element in the scene database 43 associated therewith.

20   It will be appreciated that the data stored by a job in the placeholder may be actual scene data, or it

21   may merely be status information indicating that the job has finished executing.

22   The job manager 41 makes use of a state machine to control execution of each job. The

23   current state of each job may be stored in the respective job's job description described above. The

24   job control state machine 50 used by the job manager 41 will be described in connection with FIG.

1   3. With reference to FIG. 3, in response to a request to create a job, the job control state machine

2   50 enters a job creation state 51. In the job creation state 51, the job manager creates a job

3   description, as described above, and, in addition, creates a placeholder for the scene data element(s)

4   to be associated with the job in the scene database 43. A module that issues a job creation request

5   may notify the job manager 41 of prerequisite jobs of which it is aware, either as part of the job

6   creation request or subsequent thereto, and the job manager 41 can use that information to link the

7   job in the dependency graph 44. In addition, if a job identified as a prerequisite is not already in the

8   dependency graph 44, the job manager 41 can initiate creation of the job by issuing a job creation

9   request for the job, in which case the prerequisite job will also be linked into the dependency graph

10   44. After the module that issued the job creation request has provided all of the information required

11   for the creation of the job initiated by the job creation request, including the prerequisites of which

12   it is aware, the module can issue a job creation end request, which marks the end of the job creation

13   state. At that point, the job enters a dormant state 52.

14   As noted above, a job is not executed until a request for the scene data element that would

15   be generated by the job is issued (reference item (ii) directly above). Accordingly, a job will remain

16   in the dormant state 52 until a request is made for the scene data element that would be generated

17   by the job is issued. At that point, since the scene data element is represented by a placeholder, a

18   job execute request will be generated by the scene database 43 for the job associated with the

19   placeholder, at which point the job will sequence to the pending state 53. In the pending state 53,

20   the job manager 41 will determine whether the job's prerequisite jobs have finished execution, and,

21   if so, the job will enter a runnable state 54. While a job is in the runnable state 54, the job manager

22   41 can schedule the job for execution, based on the job's estimated job cost value as described above.

23   If, while the job is in the runnable state 54, the job manager 41 identifies a new prerequisite for the

24   job, the job can return to the pending state 53 until the new prerequisite has been executed.

25   When the job manager 41 assigns the job to a thread, the job enters a running state 55. In the

26   running state, the job is actually being executed. While a job is being executed, a new prerequisite

27   may be discovered, in which case the job manager 41 will sequence the job to a suspended state 56

28   until the prerequisite can be executed. If, for some reason, an error occurs during execution of the

1   job, the job manager 41 will sequence the job to a failed state 57. If a subsequent execution request

2   is issued for the job, it will return to the pending state 53.

3   On the other hand, if the module that is executing the thread in which the job is being

4   executed, determines that the job has been successfully executed and is ready to store results in the

5   scene database 43, it will issue a job finished notification to the job manager 41. In response to the

6   job finished notification, the job manager 41 sequences the job to a finished state 58 and issues a

7   notification to the module that it can initiate a storage operation to store the results in the scene

8   database 43. The job manager 41 can control issuance of the notification allowing the module to

9   initiate the storage operation so as to control the timing with which the module can initiate the

10   storage operation. This may be desirable to maintain consistency of scene data for a scene in the

11   scene database 43 if, for example, other jobs are using the scene data for rendering. After the job

12   manager 41 has issued the notification to the module indicating that it can initiate the storage

13   operation, the module can initiate the storage operation. In one embodiment, the module will initiate

14   a storage operation, not to the placeholder(s) for the job, but to respective temporary storage

15   location(s) in the scene database 43. After the storage operation to the temporary storage location(s)

16   has completed, the module will issue a job storage done request to the job manager 41. In response

17   to the job storage done request, the job manager 41 will sequence the job to a done state 59. In

18   addition, the job manager 41 will enable the information in the temporary storage location(s) to be

19   transferred to the placeholder(s). In one embodiment, the transfer is accomplished by means of

20   updating pointers from pointing to the storage locations associated with the placeholder(s) to point

21   to the temporary storage location(s).

22   After the job manager 41 has sequenced the job to the done state 59 and updated the pointers,

23   it can sequence the job to a paging state 60, in which it will enable the job descriptor to be

24   transferred from the scene database 43 to, for example, a permanent storage arrangement (not shown)

25   such as a disk storage device. Following the paging state 60, or following the done state 59 if the

26   job is not sequenced to the paging state, a garbage collector maintained by the job manager 41 can

27   delete the job description from the scene database 43 in the manner described above, at which point

28   the job manager 41 sequences the job to a flushed state 61. If the job manager 41 subsequently

1	receives a job execution request requesting subsequent execution of the job, the job manager 41 can

2	return the job to the pending state, at which point the operations described above can be repeated.

3	The invention provides a number of advantages. In particular, the invention provides a server

4	for use in connection with one or more clients that can render images of scenes for delivery to

5	respective clients. The server can render images of the same scene for a plurality of the clients, and

6	can allow the clients to request customizations of the scene and control those clients for which the

7	customizations will be visible in images rendered therefor. In particular, the server can selectively

8	allow customizations requested by one client to be made in a scene for which images are rendered

9	for, for example, only that client, for a selected group of contemporaneously-connected clients, or

10	for all contemporaneously-connected clients for which images are to be rendered of the scene.

11	By rendering the images itself, instead of providing three-dimensional scene information to

12	the respective clients, the server ensures that confidential three-dimensional scene information

13	concerning objects in a scene is not provided to the respective clients.

14	In addition, by allowing images of customized scene to be provided, not only to the client

15	that requested the customization, but also to all or a selected subset of the other clients that are

16	engaged in a session involving the same scene, the server also manages cooperative or competitive

17	efforts by a plurality of users using respective ones of the client, allowing, for example, a plurality

18	of users in diverse locations using respective client devices to cooperatively design a product, a

19	plurality of users in diverse locations to play games, and other efforts as will be apparent to those

20	skilled in the art.

21	In addition, since rendering is performed by the server, and not by the clients, the

22	arrangement allows clients to be used that have relatively limited storage and processing capacities,

23	such as Web pads, personal data assistants (PDA's), cellular telephones and the like, and still allow

24	for arbitrarily complex manipulation of three-dimensional scene data.

25	It will be appreciated that numerous changes and modifications may be made to the

26	arrangement as described herein. For example, although the server has been described as rendering

1   images in response to image retrieval requests issued by respective clients, it will be appreciated that

2   the server can render images in response to other events. For example, the server can transmit an

3   updated image to one client in response to an event initiated by another client. This can be useful

4   if, for example, users of the respective clients are engaged in a collaborative design effort, playing

5   a game, or the like. In addition, the server can stream updated images to one or more clients if the

6   images are to represent, for example, video, using any convenient image transfer or streaming video

7   transfer protocol. Furthermore, although the server has been described as providing images rendered

8   in response to requests issued by browsers in connection with links associated with Web pages, it

9   will be appreciated that the image may be rendered and provided in response to requests issued by

10   other types of programs executed by the client devices, in connection with other types of request

11   initiation mechanisms, which need not be associated with a Web page.

12   It will be appreciated that tools through which a user may request customizations to a scene

13   may be implemented using any type of program. If customizations are requested through Web

14   pages, as described above, the tools may be efficiently implemented by means of, for example,

15   applets provided with the Web pages, which applets may be in, for example, the well-known Java

16   programming language.

17   It will further be appreciated that a server may be implemented on a single processing

18   platform, or on multiple processing platforms. The scene database 43, for example, may be

19   implemented as a virtual shared database that is distributed across a plurality of processing

20   platforms, and various components of the server may also be distributed across the various

21   processing platforms.

22   In addition, although specific arrangements have been described by which the job manager

23   41 determines which jobs are to be executed, determines job cost values and deletes scene data

24   elements from the scene database 43, it will be appreciated that other arrangements can be used.

25   Furthermore, it will be appreciated that, although a specific state machine has been described by

26   which the job manager controls execution of a job (reference FIG. 3), it will be appreciated that other

27   state machines, and arrangements other than state machines, may be used instead.

1   It will be appreciated that a system in accordance with the invention can be constructed in

2   whole or in part from special purpose hardware or a general purpose computer system, or any

3   combination thereof, any portion of which may be controlled by a suitable program. Any program

4   may in whole or in part comprise part of or be stored on the system in a conventional manner, or it

5   may in whole or in part be provided in to the system over a network or other mechanism for

6   transferring information in a conventional manner. In addition, it will be appreciated that the system

7   may be operated and/or otherwise controlled by means of information provided by a user using user

8   input elements (not shown) which may be connected directly to the system or which may transfer

9   the information to the system over a network or other mechanism for transferring information in a

10   conventional manner.

11   The foregoing description has been limited to a specific embodiment of this invention. It will

12   be apparent, however, that various variations and customizations may be made to the invention, with

13   the attainment of some or all of the advantages of the invention. It is the object of the appended

14   claims to cover these and such other variations and customizations as come within the true spirit and

15   scope of the invention.

16   What is claimed as new and desired to be secured by Letters Patent of the United States is: